
Robot Programming #2

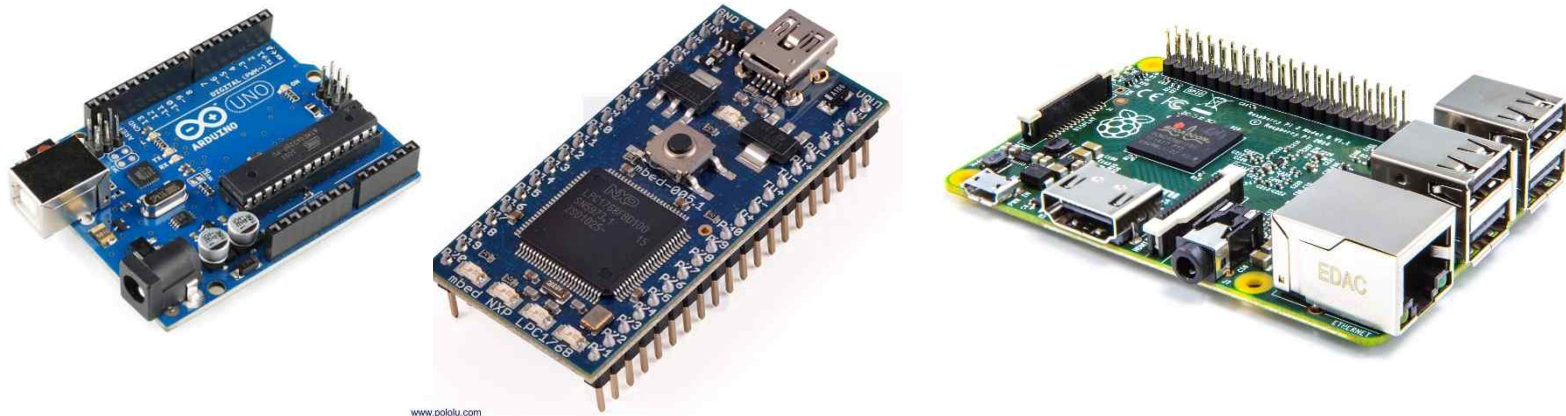
로봇제어기 - 아두이노 우노

**Dept. of Mech. Robotics and Energy Eng.
Dongguk University**



How to control robots?

- We need a controller.
- The controller controls inputs and outputs according to program.
- There are numerous controllers available in the market.

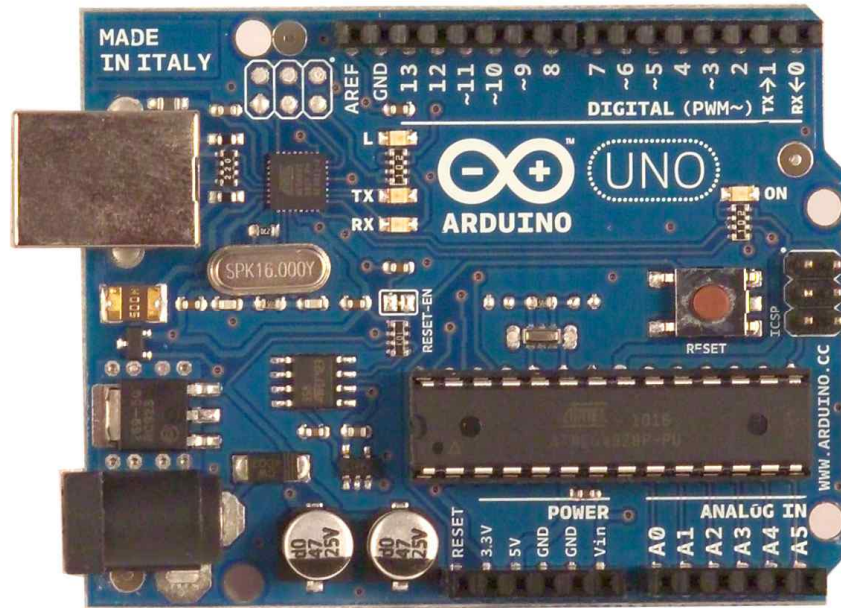


During experiment

- Don't bring food or beverage to the class.
- Turn off the power S/W before connecting or disconnecting cable!!!!
- When you bread-boarding, turn off power always!!!!
- Don't apply overload to the system.
- Electronic parts are volatile. Be careful when you handle them.
- Safety is first.

What is an Arduino?

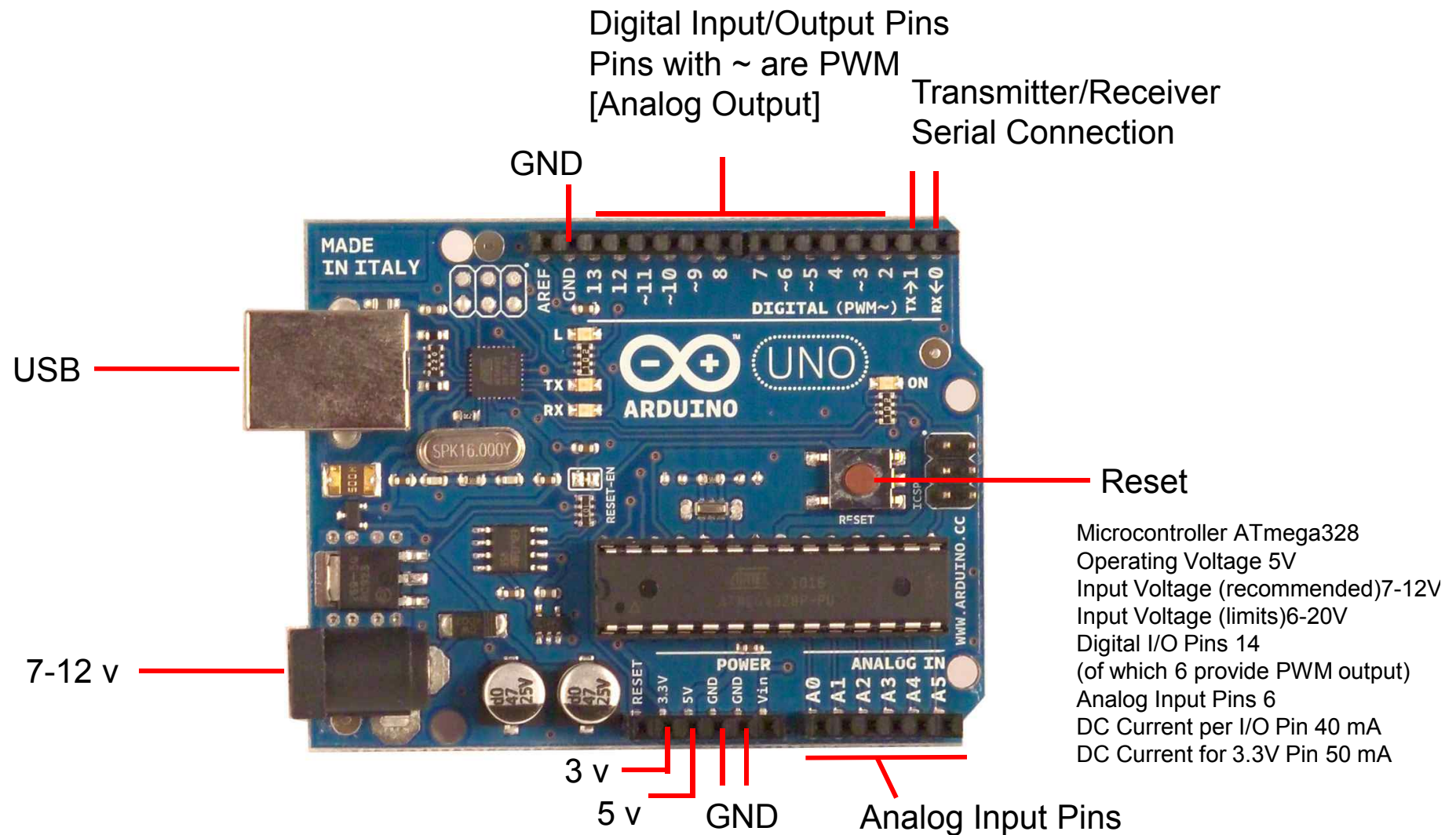
- Open source platform suitable to prototyping.
- Hardware connections and software are easy to use.



Getting Started with arduino controller

- To use Arduino controller, you need
 1. PC and Internet connection
 2. USB port
- Arduino is designed for beginners with little knowledge on electronics and microcontrollers.
- The advantage of the Arduino is that it doesn't require complex concepts such as flag and register.
- Basic commands such as **digitalWrite** are used to control ports.

Arduino Uno

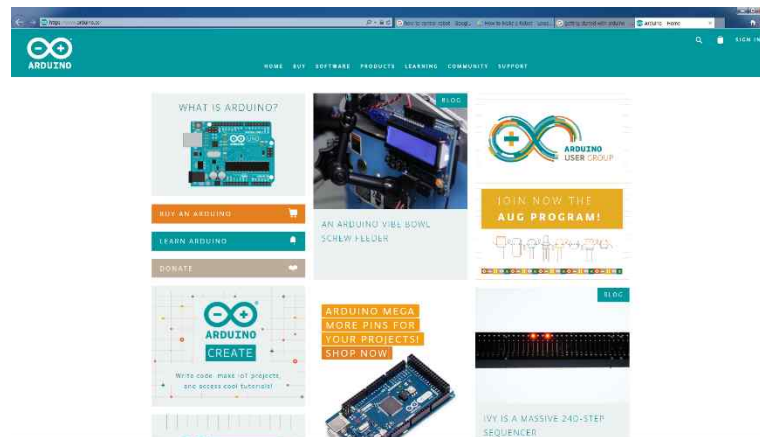


Arduino Uno

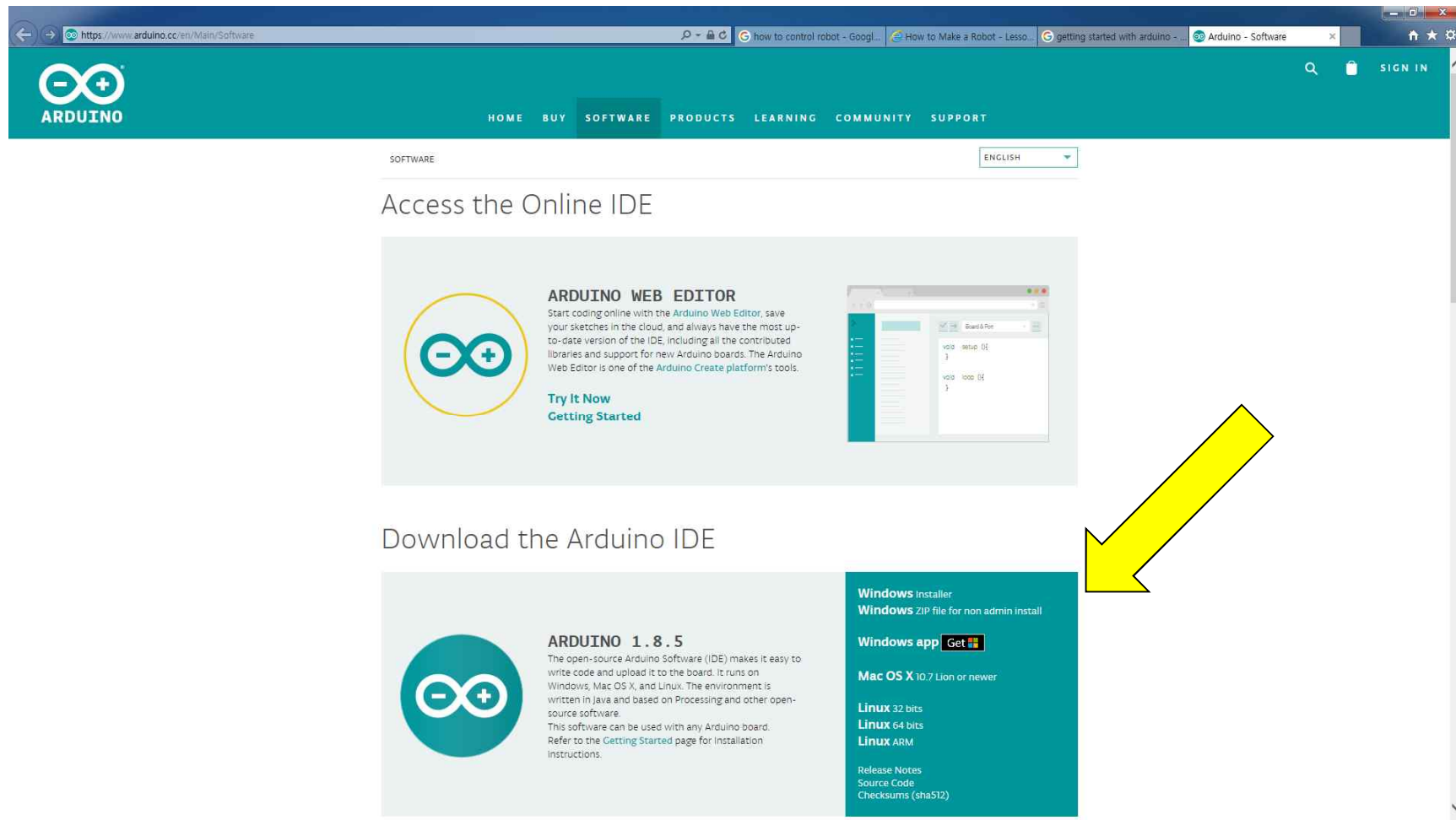
- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- 14 Digital I/O Pins (of which 6 provide PWM output)
- 6 Analog Input Pins
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA

Download Arduino software



- Download Arduino Software from the website [Arduino.cc](https://www.arduino.cc) and unzip the folder to your computer. A file within the folder called Arduino, allows you to launch the programming environment.
- You need to install a driver that comes with Arduino to be able to communicate with the board.




Download the Arduino IDE



The screenshot shows the Arduino Software page with the following content:

- Access the Online IDE**
 - ARDUINO WEB EDITOR**: Start coding online with the Arduino Web Editor, save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards. The Arduino Web Editor is one of the Arduino Create platform's tools.
 - [Try It Now](#)
 - [Getting Started](#)
 - 
- Download the Arduino IDE**
 - **ARDUINO 1.8.5**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.
 - Windows** [Installer](#)
[Windows ZIP file for non admin install](#)
Windows app [Get](#) 
Mac OS X 10.7 Lion or newer
Linux 32 bits
Linux 64 bits
Linux ARM
[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

A large yellow arrow points from the 'Download the Arduino IDE' section to the 'Windows Installer' link.

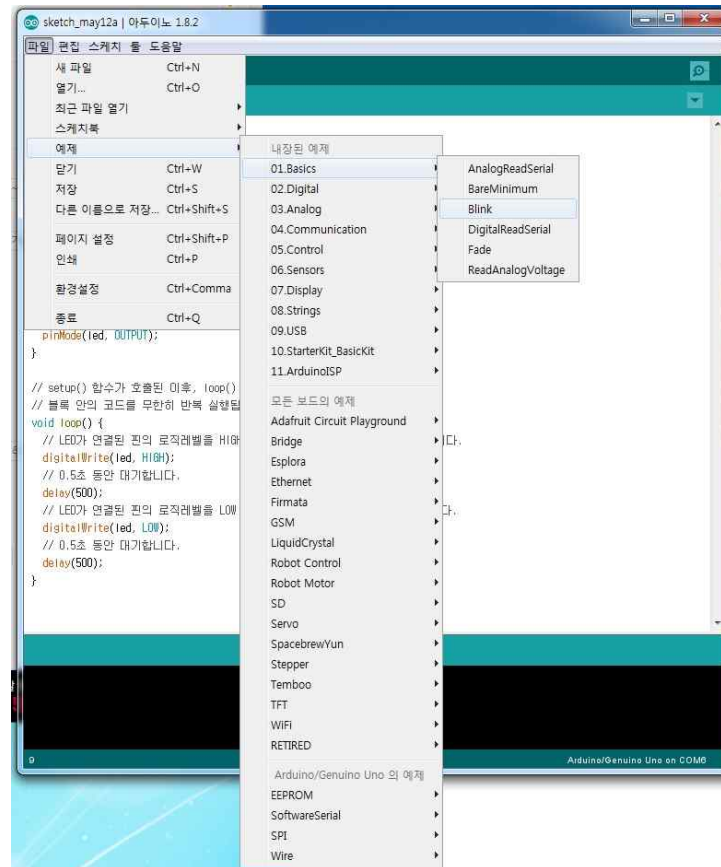
Getting Started with Arduino

- Let us connect the Arduino to a PC as shown below.



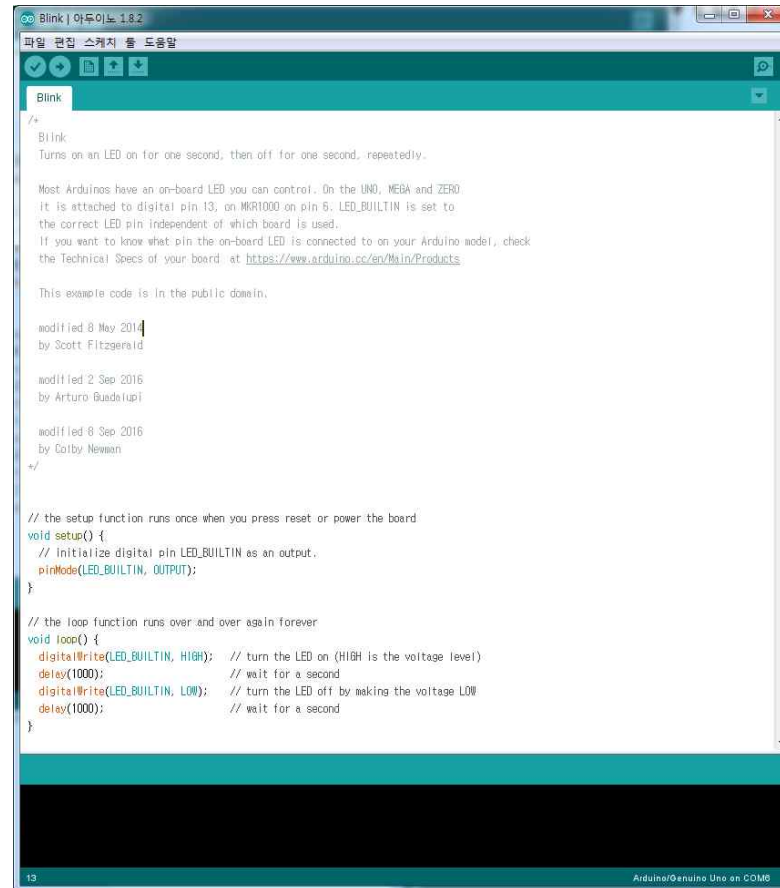
Getting Started with Arduino IDE

- Run Arduino IDE and open Blink example.



Getting Started with Arduino IDE

- Blink example



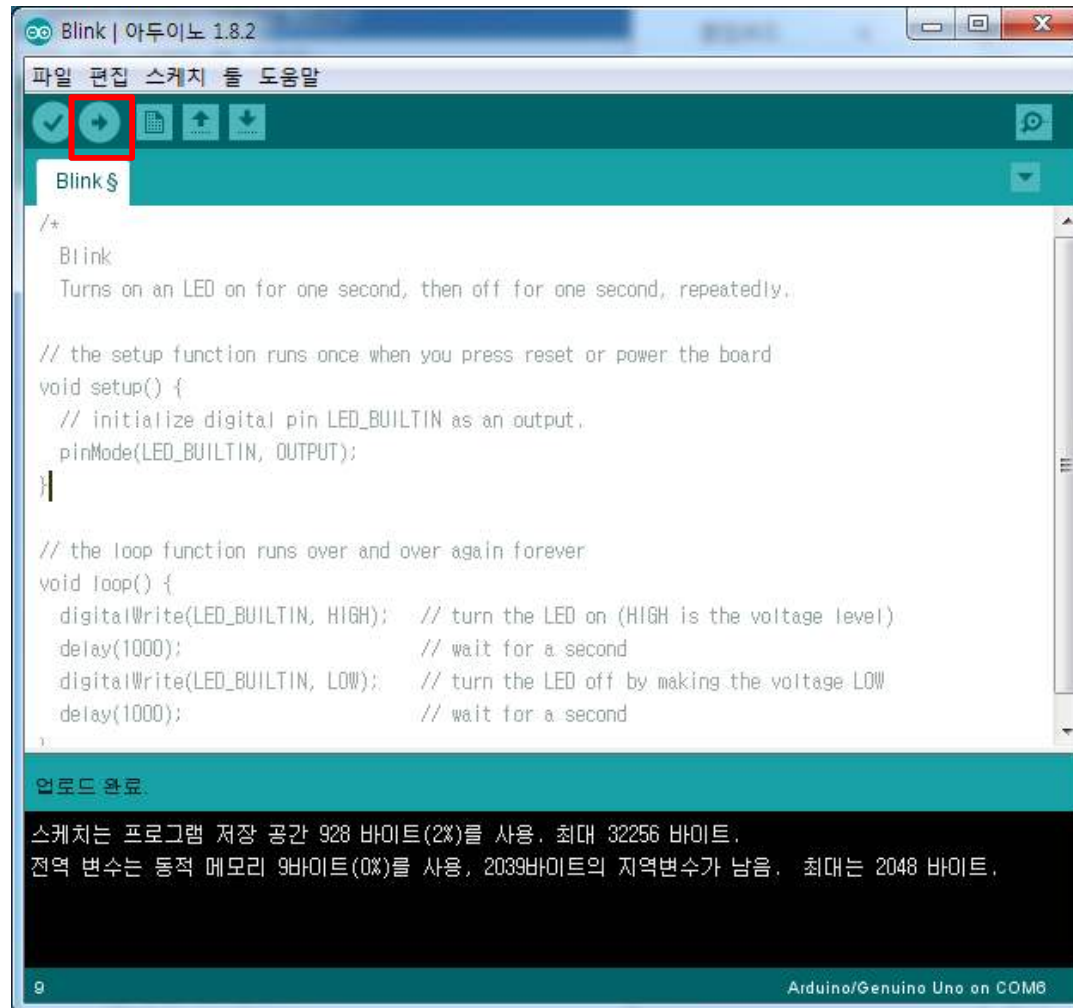
The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The code is written in C++ and is designed to turn an on-board LED on and off in a repeating cycle. The code includes comments explaining the setup and loop functions, as well as the specific pin and delay values used. The IDE window title is 'Blink | 아두이노 1.8.2'. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. (On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on M1000 to pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino model, check
 * the Technical Specs of your board at: https://www.arduino.cc/en/Main/Products
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 *
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 *
 * modified 8 Sep 2016
 * by Colby Newman
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

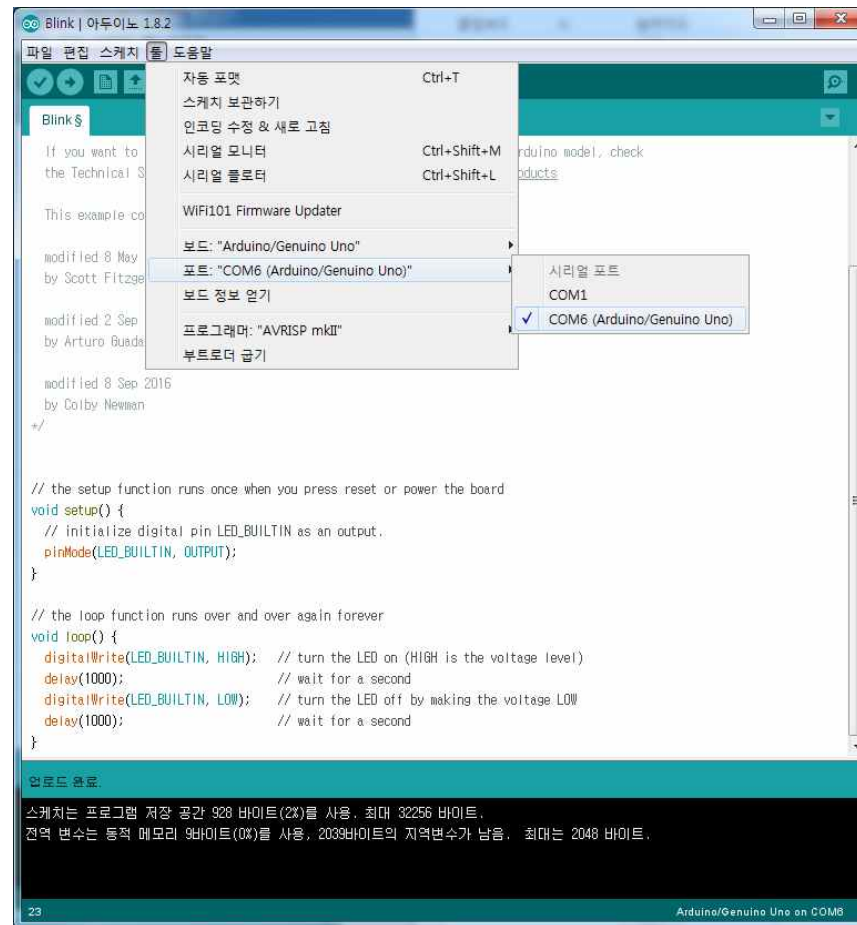
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Getting Started with Arduino IDE

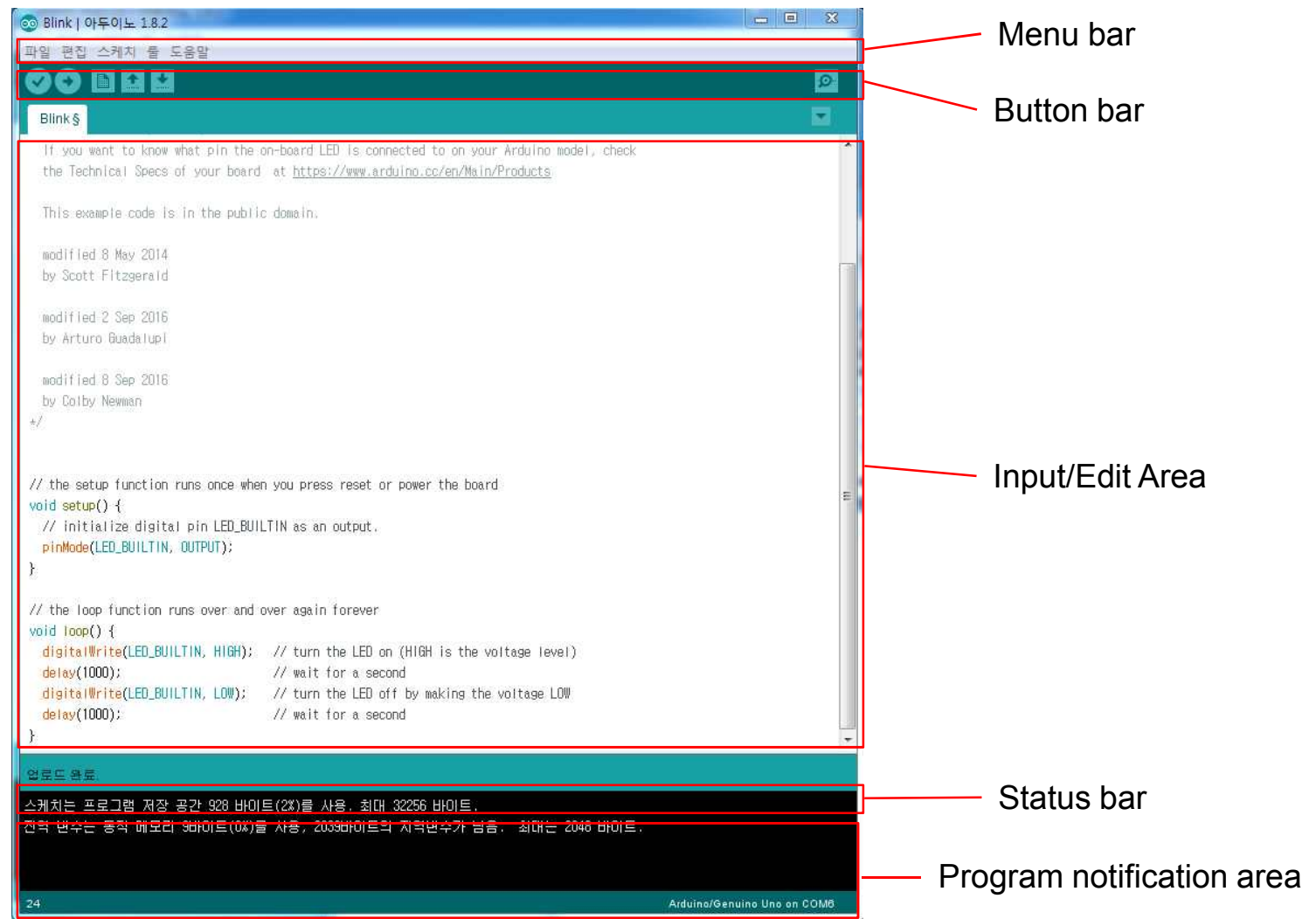


Getting Started with Arduino IDE

- Select serial port and board



Getting Started with Arduino IDE



Getting Started with Arduino IDE

- Press compile button to check for errors.
- Press upload button to program Arduino board with your sketch.
- What happens to the Arduino?

Getting Started with Arduino IDE

- View the default program source code.
- Can you tell me what is going on?

Getting Started with Arduino IDE

- Let's take a look at the Blink program.
- Bare minimum code consists of

```
void setup() {  
  // put your setup code here, to run once.  
}  
  
void loop() {  
  // put your main code here, to run repeatedly.  
}
```

Getting Started with Arduino IDE

- **setup:** This is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes.
- **loop:** The loop function runs repeatedly until the device is powered off. The main logic of the code enters here. Similar to while(1) in other microcontroller programming.

Getting Started with Arduino IDE

- A pin on the Arduino board can be set as either input or output by using the `pinMode` function.
- `pinMode(LED_BUILTIN,OUTPUT)` implies that `LED_BUILTIN` pin is set as output pin.
- The `LED_BUILTIN` is a small LED on the board. By default its pin number is 13.

Getting Started with Arduino IDE

- We can set the digital value of the pin by using `digitalWrite`.
- `digitalWrite(LED_BUILTIN,HIGH)`: it makes the output voltage of `LED_BUILTIN`(pin 13) 5V. It turns the LED on (HIGH is the voltage level).
- `digitalWrite(LED_BUILTIN,LOW)`: it makes the output voltage of `LED_BUILTIN`(pin 13) 0V. It turns the LED off.
- `delay(1000)`: Arduino will wait for a second(1000ms).